

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s) : Heiko DASSOW et al.
Serial No. : 09/913,371
Filed : August 13, 2001
For : METHOD FOR THE GRAPHIC REPRESENTATION
AND/OR PROCESSING VALUES OF DATA TYPES

Art Unit : 2174
Confirmation No. : 9518
Examiner : Boris M. Pesin

Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I hereby certify that this correspondence is being electronically
transmitted to the United States Patent and Trademark Office via the
Office electronic filing system on

Date: **October 8, 2008**

Signature: Linda Lecomte/
Linda Shudy Lecomte, Reg. No. 47,084

AMENDMENT

Sir:

In response to the Office Action made final after an RCE filing, please reconsider the
above-identified application based on the following.

Amendments to the Claims begin on page 2 of this paper.

Remarks begin on page 6 of this paper.

AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

Claims 1 to 12. (Canceled).

13. (Previously Presented) A method for graphically representing a value of a data type of a formally defined data structure existing as a value tree, comprising:

assigning a window as a graphical user interface to the data structure;

inserting hierarchically at least one generic, scalable, graphical user-interface component in the window, the value tree of the data structure being mapped onto the at least one user-interface component;

providing that the at least one graphical user interface component is in a recognizable relation to at least one node of the value tree;

providing at least one of a graphical representation and a textual representation of the value is selectable for each subtree of the value tree; and

for a processing of the value tree, deriving for each node a value list of all of values compatible with respect to assignment with the data types, and selecting one of the value from the value list for each value assignment; wherein, when compiling the value list, the number of the values to be accepted in the list being restricted in accordance with predefined rules depending on the current context.

14. (Canceled).

15. (Canceled).

16. (Previously Presented) The method of claim 13, wherein a visualization of the window is first undertaken at a time of an initialization of the graphical user-interface and, after that, at least one of data and the value list is initialized, which are derived for a processing.

17. (Previously Presented) The method of claim 13, wherein the value to be represented is transferred in a transfer syntax containing all necessary information for the representation with respect to the data type and the value assignment.

18. (Previously Presented) The method of claim 13, wherein the data type, whose exact type assignment can first be determined at execution time in accordance with a late binding

principle, is inserted as a dynamically changeable subtree in the value tree represented by the graphical user-interface.

19. (Previously Presented) The method of claim 13, wherein for the data type whose exact type assignment is first defined in accordance with a late binding principle at an execution time by a marking of another node, a user is prompted to input information as to whether the exact type assignment should be performed one of automatically and following a manual input.

20. (Previously Presented) The method of claim 13, wherein the value can be transferred from the subtree to another subtree by intermediately storing and clicking on the subtree.

21. (Previously Presented) The method of claim 13, wherein the method is implemented by at least one program module that is integratable in an application program.

22. (Previously Presented) The method of claim 13, wherein additional information to be displayed is storable for each of the at least one node of the value tree which can be uniquely named by a displayed type and a relation to the higher-level type.

23. (Previously Presented) The method of claim 13, further comprising:
continually checking during an inputting of the value of the data type in the value tree to determine whether an input value is permissible for a corresponding data type and to determine whether the input value is identical to a currently active value of the corresponding data type; and
making known to a user a result of the continually checking.

24. (Previously Presented) The method of claim 23, wherein a display format is alterable when the value is inputted before the value is accepted into the value tree.

25. (Previously Presented) The method of claim 19, wherein the marking of the another node includes "ANY DEFINED BY" in a description language ASN.1.

26. (Previously Presented) The method of claim 24, wherein a numerical value is displayed as one of a decimal value and a binary value.

27. (Previously Presented) A method for graphically representing a value of a data type of a formally defined data structure existing as a value tree, comprising:
assigning a window as a graphical user interface to the data structure;

inserting hierarchically at least one generic, scalable, graphical user-interface component in the window, the value tree of the data structure being mapped onto the at least one user-interface component;
providing that the at least one graphical user interface component is in a recognizable relation to at least one node of the value tree;
providing at least one of a graphical representation and a textual representation of the value is selectable for each subtree of the value tree; and

for a processing of the value tree, deriving for each node a value list of all of values compatible with respect to assignment with the data types, and selecting one of the value from the value list for each value assignment,

wherein, when compiling the value list, the number of the values to be accepted in the list is restricted in accordance with predefined rules depending on the current context, and a visualization of the window is first undertaken at a time of an initialization of the graphical user-interface and, after that, at least one of data and the value list is initialized, which are derived for a processing.

28. (Previously Presented) The method of claim 27, further comprising:

continually checking during an inputting of the value of the data type in the value tree to determine whether an input value is permissible for a corresponding data type and to determine whether the input value is identical to a currently active value of the corresponding data type; and

making known to a user a result of the continually checking.

29. (Previously Presented) The method of claim 27, wherein the value to be represented is transferred in a transfer syntax containing all necessary information for the representation with respect to the data type and the value assignment.

30. (Previously Presented) The method of claim 29, wherein the data type, whose exact type assignment can first be determined at execution time in accordance with a late binding principle, is inserted as a dynamically changeable subtree in the value tree represented by the graphical user-interface.

31. (Previously Presented) The method of claim 29, wherein for the data type whose exact type assignment is first defined in accordance with a late binding principle at an execution time by a marking of another node, a user is prompted to input information as to whether the exact type assignment should be performed one of automatically and following a manual input.

32. (Previously Presented) The method of claim 29, wherein additional information to be displayed is storable for each of the at least one node of the value tree which can be uniquely named by a displayed type and a relation to the higher-level type.

33. (New) The method of claim 16, wherein the value to be represented is transferred in a transfer syntax containing all necessary information for the representation with respect to the data type and the value assignment, the value transferable from the subtree to another subtree by intermediately storing and clicking on the subtree, and wherein the data type, whose exact type assignment can first be determined at execution time in accordance with a late binding principle, is inserted as a dynamically changeable subtree in the value tree represented by the graphical user-interface, and wherein additional information to be displayed is storable for each of the at least one node of the value tree which can be uniquely named by a displayed type and a relation to the higher-level type.

34. (New) The method of claim 33, further comprising:

continually checking during an inputting of the value of the data type in the value tree to determine whether an input value is permissible for a corresponding data type and to determine whether the input value is identical to a currently active value of the corresponding data type; and
making known to a user a result of the continually checking.

35. (New) The method of claim 16, wherein for the data type whose exact type assignment is first defined in accordance with a late binding principle at an execution time by a marking of another node, a user is prompted to input information as to whether the exact type assignment should be performed one of automatically and following a manual input.

REMARKS

Claims 33 to 35 are new. No new matter has been added. Claims 13 and 16 to 32 are pending in the present application. Applicants request reconsideration of the present application in view of this response.

35 U.S.C. § 103(a) – Ebert Reference, Crim Reference

Claims 13, 15 to 18, 20 to 24, and 26 to 32, were rejected under 35 U.S.C. § 103(a) as unpatentable by U.S. Patent No. 6,278,991 to Ebert in view of U.S. Patent No. 5,920,866 to Crim (“Crim reference”).

Applicants respectfully submit that the Ebert reference taken in combination with the Crim reference is not believed to teach or suggest each and every feature of the claims.

The Ebert reference, at col. 2 as cited by the Office Action, recites merely that it is “an object of this invention to convey data in an efficient manner by displaying the data according to the data’s hierarchical structure.” Further, the Ebert reference, at Fig. 6a and its accompanying text at col. 7, lines 48-55, also cited by the Office Action, recites the situation in which a user selects object 522 (category “Financial Situation”), the result is a screen 610 showing lens 618, which is displaying a graphical representation of the category “Financial Situation.” The Ebert reference further recites that the lens 618 displaying a graphical representation of object 620 that is a category one hierarchy level lower than “Financial Situation,” and graphical representations of objects 630, 632, and 634, which are not categories.

These two passages do not expressly disclose the requirements of amended method claim 13, including *for a processing of the value tree, deriving for each node a value list of all of values compatible with respect to assignment with the data types, and selecting one of the value from the value list for each value assignment*, in the manner claimed. Further, the Abstract of the references also does not provide such disclosure, in contrast to the Office Action’s assertions. Instead, the Ebert reference appears to concern the hierarchical structuring of data by providing access to data at several levels within the hierarchical data structure simultaneously. Further, claim 13 also requires that *when compiling the value list, the number of the values to be accepted in the list being restricted in accordance with predefined rules depending on the current context*.

The Crim reference does not cure the deficiencies of the Ebert reference. The Crim reference appears to concern a process and apparatus for generating and editing value lists. While the Crim reference refers to a user creating a value list having values selected from

other fields, the created value list is given an identifying name so that it may be shared by other fields in a current database or another database. This is different than the value lists and rules employed by the claims of the present invention.

Even if one of ordinary skill in the art combined the two references, one would not necessarily obtain the invention of claim 13 in the present application. That is, while the Ebert reference appears to concern itself with links or data in hierarchical situations, and the Crim reference appears to concern itself with generating and editing value lists, ***it does not necessarily follow that one of ordinary skill in the art would consider deriving for each node a value list of all values compatible with respect to assignment with the data types, and selecting one of the values from the values list for each value assignment, as in claim 13, among other features.*** For motivation, the prior art references must suggest the desirability of such a combination, and the prior art references need to concern a similar “nature of the problem.” In re Mills, 16 USPQ2d 1430 (Fed. Cir. 1990). Applicants respectfully submit, as discussed in detail above, that the Ebert and Crim references concern different issues and do not suggest – either expressly or implicitly – the desirability of such a combination.

In addition, it would be also improper to later suggest that the claimed features of a visualization of the window is first undertaken at a time on initialization *and* implementing specific rules when creating a value list are inherent in the claimed references, as well as suggesting that the same references are properly combinable. To suggest such inherency, there will need to be provided a “basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristics *necessarily* flows from the teachings of the applied art.” (See M.P.E.P. § 2112; emphasis in original; and *see Ex parte Levy*, 17 U.S.P.Q.2d 1461, 1464 (Bd. Pat. App. & Int’f. 1990)). Thus, the M.P.E.P. and the case law make clear that simply because a certain result or characteristic may occur in the prior art does not establish the inherency of that result or characteristic.

Applicants respectfully submit that the Ebert and Crim references do not render obvious amended claim 13 or its dependent claims 15 to 18, 20 to 24, and 26. Further claim 27 and its dependent claims 28 to 32 recite features analogous to those of claim 13 and are allowable for essentially the same reasons. Accordingly, Applicants respectfully submit that claims 13, 15 to 18, 20 to 24, and 26 to 32, are allowable under 35 U.S.C. § 103(a) over the Ebert reference in view of the Crim reference.

35 U.S.C. § 103(a) – Ebert Reference, Crim Reference, Gardner Reference

Claims 19 and 25 were rejected under 35 U.S.C. § 103(a) as anticipated by the Ebert reference in view of the Crim reference and further in view of U.S. Patent No. 6,701,352 to Gardner (“Gardner reference”).

Claims 19 and 25 depend from claim 13. As discussed above, claim 13 is believed allowable over the Ebert and Crim references (taken alone or in combination).

The Gardner reference does not appear to cure the deficiencies of the Ebert and Crim references. The Gardner reference appears to concern automatically importing, according to predefined selection criteria, information stored at a resource remotely located across a network. The Gardner reference, even when taken in combination with the Ebert reference, does not teach or suggest at least the feature of for a processing of the value tree, deriving for each node a value list of all of values compatible with respect to assignment with the data types, and selecting one of the value from the value list for each value assignment as in claim 13, and thus, as in dependent claims 19 and 25. Accordingly, Applicants respectfully submit that claims 19 and 25 are allowable under 35 U.S.C. § 103(a) over the Ebert reference in view of the Crim reference and further in view of the Gardner reference.

In summary, it is respectfully submitted that all of claims 13 and 16 to 32 of the above-identified application are allowable for at least the foregoing reasons.

CONCLUSION

In view of the foregoing, it is respectfully submitted that all claims 13 and 16 to 35, as presented above, are allowable. It is therefore respectfully requested that the present application issue as early as possible.

Respectfully submitted,

Dated: October 8, 2008

CUSTOMER NO. 26646

By: /Linda Lecomte/
Linda Shudy Lecomte (Reg. No. 47,084)
KENYON & KENYON LLP
One Broadway
New York, New York 10004
(212) 425-7200